UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,722	11/09/2000	Stephan J. Jourdan	2207/9800	2194

25693 7590 08/18/2009

KENYON & KENYON LLP
RIVERPARK TOWERS, SUITE 600
333 W. SAN CARLOS ST.
SAN JOSE, CA 95110

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
08/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/708,722	JOURDAN ET AL.	
	Examiner	Art Unit	
	AIMEE J. LI	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 May 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-19 have been considered.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 18 May 2009.

Response to Arguments

3. Applicant's arguments filed 18 May 2009 have been fully considered but they are not persuasive. Applicants' argue in essence on pages 7-10

> Applicants submit the cited references do not teach, suggest or disclose at least "[a] cache comprising: a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line in reverse program order, wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line"...

4. This has not been found persuasive. In the rejection below, the cited sections of Peleg teaches the limitations of the claims. All sections cited show the claim limitations and are needed to be read together, in context, to fully understand the teachings of Peleg. The following quotations to show what language in Peleg was associated with the claim limitations are just examples of the teachings in Peleg.

5. Peleg teaches "a cache line to store an instruction segment" in column 8, lines 41-49, which states "…The instructions contained in the line buffer **22** and their transfer to the cache memory **23** for the trace...", and in Figure 5B teaches storing the basic instruction blocks in a buffer, i.e. cache, line in the order shown in Figure 5B. Peleg further shows in Figures 6-12 and

associated description, such as column 9, lines 36-51, the storage of basic instruction blocks in buffer lines 73, 74, and 75 to a line in cache memory.

6. Peleg teaches "an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line in reverse program order" in column 8, lines 30-43 and Figure 5B, where is teaches that the basic instruction blocks can execute in reverse order by executing basic blocks 1-4 and then executing basic block 2 based upon the exit branching of basic block 4. Peleg shows how these basic blocks are stored in the cache lines in Figure 6-12. Peleg specifically shows in Figure 10, element 73 that basic block 4 is stored first in the line, then basic block 2 is stored in the second position in the line.

7. Peleg teaches "wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line" in the various sited sections in the rejection below. Peleg first teaches in column 4, lines 11-23 that a "basic block comprises instructions in a computer program which are unconditionally and consecutively executed." The first instruction in the basic block is "the first instruction following a branch instruction and its last instruction is a branch instruction. "Branch instructions include both unconditional branches (e.g., call, return) and *conditional branches (emphasis added)...*" This means that the basic blocks end with a conditional branch, which changes the program flow by jumping from one block of instructions to another. For example, Figures 5A and 5B shows the order of execution of a group of basic blocks, and it can be seen that the exiting branches, which includes conditional branches, can change program flow. A seen in Figure 5A, BB1 begins at A1, then BB2 begins at A12, then BB3 goes backwards and

begins at A3, etc. Figure 5A also shows that BB5 has only a single instruction, since it beings and ends at A24. This means that the instruction at A24 in BB5 is the beginning and ending, i.e. terminal, instruction for BB5. So, when this BB5 is stored, this single instruction is stored. As shown in Figure 11, element 75 and column 10, lines 25-42, BB5 is stored in the first location in buffer line 75 when it is executed as a result of the exit jump from BB2.

8. Examiner notes that Applicant states on page 8, paragraph 2 "…Applicants submit that because of the extensive nature of the section...Applicants respectfully request further clarification of the current rejection by citation to a more specific portion of the cited section." The Examiner is unclear as to why this section has an "extensive nature". The cited sections in the previous rejection, copied below, were all deemed necessary to fully understand Peleg. The Examiner would also point out that, including column 8, line 30 to column 9, line 35, a total 167 lines were cited out by the Examiner out of Peleg's total 893 lines, which is only about 19% of Peleg. So, the Examiner is unsure about the "extensive nature" of the citations. Sometimes it is necessary to cite an entire column or two for the functionality and features relied upon in a reference to be fully understood. If there is uncertainty or confusion as to why a section is cited, the Examiner encourages Applicants or Applicants' representative to contact the Examiner directly.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 1-5, 8, and 11-14 are rejected under 35 U.S.C. 102(b) as being taught by Peleg et al., U.S. Patent Number 5,381,533 (herein referred to as Peleg). Examiner notes that this patent was cited by Applicants in the IDS filed 29 December 2000.

11. Referring to claim 1, Peleg has taught a cache comprising:

a. a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of cache line in reverse program order (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, the cache stores the basic blocks in the order they are executed, including, as shown in Figure 10, when the basic blocks go backwards, i.e. reverse order.),

b. wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, a basic block starts with the instruction executed after a branch and terminates with a branch instructions. This means that when a basic block's first instruction is a branch instruction, it terminates with that instruction, and the basic block only consists of the one branch instruction. So, when the single

instruction basic block is jumped to, that single beginning and ending instruction is stored in the cache.).

12. Referring to claim 2, Peleg has taught the cache of claim 1, wherein the instruction segment is an extended block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

13. Referring to claim 3, Peleg has taught the cache of claim 1, wherein the instruction segment is a trace (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

14. Referring to claim 4, Peleg has taught the cache of claim 1, wherein the instruction segment is a basic block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

15. Referring to claim 5, Peleg has taught a segment cache for a front-end system in a processor, comprising

a. a plurality of cache entries to store instructions of instruction segments in reverse program order (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, the cache stores the basic blocks in the order they are executed, including, as shown in Figure 10, when the basic blocks go backwards, i.e. reverse order.),

b. wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, a basic block starts with the instruction executed after a branch and terminates with a branch instructions. This means that when a basic block's first instruction is a branch instruction, it terminates with that instruction, and the basic block only consists of the one branch instruction. So, when the single instruction basic block is jumped to, that single beginning and ending instruction is stored in the cache.).

16. Referring to claim 8, Peleg has taught a method comprising:

a. building an instruction segment based on program flow (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12), and

b. storing instructions of the instruction segment in a cache entry in reverse program order (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg,

the cache stores the basic blocks in the order they are executed, including, as shown in Figure 10, when the basic blocks go backwards, i.e. reverse order.),

c. wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, a basic block starts with the instruction executed after a branch and terminates with a branch instructions. This means that when a basic block's first instruction is a branch instruction, it terminates with that instruction, and the basic block only consists of the one branch instruction. So, when the single instruction basic block is jumped to, that single beginning and ending instruction is stored in the cache.).

17. Referring to claim 11, Peleg has taught the method of claim 8, wherein the instruction segment is an extended block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

18. Referring to claim 12, Peleg has taught the method of claim 8, wherein the instruction segment is a trace (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

19. Referring to claim 13, Peleg has taught t he method of claim 8, wherein the instruction segment is a basic block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

20. Referring to claim 14, Peleg has taught a processing engine, comprising:

a. a front end stage to build and store instruction segments, instructions provided therein in reverse program order (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, the cache stores the basic blocks in the order they are executed, including, as shown in Figure 10, when the basic blocks go backwards, i.e. reverse order.),

b. wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, a basic block starts with the instruction executed after a branch and terminates with a branch instructions. This means that when a basic block's first instruction is a branch instruction, it terminates with that instruction, and the basic block only consists of the one branch instruction. So, when the single

instruction basic block is jumped to, that single beginning and ending instruction is stored in the cache.), and

c. an execution unit in communication with the front end stage. (Peleg Figure 1).

Claim Rejections - 35 USC § 103

21. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22. Claims 6-7 and 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peleg et al., U.S. Patent Number 5,381,533 (herein referred to as Peleg) as applied to claims 5 and 14 above, and further in view of Rotenberg et al.'s "A Trace Cache Microarchitecture and Evaluation" IEEE ©1999 (herein referred to as Rotenberg).

23. Referring to claim 6, Peleg has taught the segment cache of claim 5, further comprising:

a. an instruction cache system (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12),

b. an instruction segment system, comprising:

i. a fill unit provided in communication with the instruction cache system (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

24. Peleg has not taught a selector coupled to an output of the instruction cache system and to an output of the segment cache. Rotenberg has taught a selector coupled to an output of the instruction cache system and to an output of the segment cache (Rotenberg Section 2.1 Trace-Level Sequencing "…The output of the trace cache is one or more traces…"; Section 2.2 Instruction-Level Sequencing "The *outstanding trace buffers* in Fig. 2 are used to 1) construct new traces that are not in the trace cache and 2) track branch outcomes…"; and Figure 2 – In regards to Rotenberg, Figure 2 shows the output of the trace cache and the outstanding trace buffers are connected to a line that has two inputs and one output, which is a selector. The selector chooses between and existing trace in the trace cache, i.e. a trace cache hit, or a newly formed trace, i.e. a trace cache miss or misprediction.) A person of ordinary skill in the art at the time the invention was made, and as taught by Rotenberg, would have recognized that the trace cache system provides fast trace-level sequencing while providing a method to create nonexistent traces or repair mispredicted traces (Rotenberg Section 2.1 Trace-Level Sequencing "…The trace predictor and trace cache together provide fast trace-level sequencing…Instruction-level sequencing, discussed in the next section, is required to construct nonexistent traces or repair trace mispredictions."). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the trace caches, buffers, and selector of Rotenberg in the device of Peleg to increase the speed of trace-level sequencing while using instruction-level sequencing to create new or correct mispredicted traces.

25. Referring to claim 7, Peleg has not taught an apparatus of claim 6, wherein the instruction segment system further comprises a segment predictor provided in communication with the segment cache. Rotenberg has taught an apparatus of claim 6, wherein the instruction segment

system further comprises a segment predictor provided in communication with the segment cache (Rotenberg Section 2.1 Trace-Level Sequencing "…A *next trace predictor*[14] treats traces as basic units and explicitly predicts sequences of traces…" and Figures 2). A person of ordinary skill in the art at the time the invention was made, and as taught by Rotenberg, would have recognized that the trace predictor achieves high branch prediction throughput with a single prediction per cycle (Rotenberg Section 2.1 Trace-Level Sequencing "…high branch prediction throughput is implicitly achieved with only a single trace prediction per cycle…"). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the trace predictor of Rotenberg in the device of Peleg to achieve high branch prediction throughput in a single prediction cycle.

26. Referring to claim 15, Peleg has taught the processing engine of claim 14, wherein the front-end stage comprises:

a. an instruction cache system (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12),

b. an instruction segment system, comprising:

ii. a fill unit provided in communication with the instruction cache system (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12),

iii. a segment cache (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

27. Peleg has not taught a selector coupled to an output of the instruction cache system and to an output of the segment cache. Rotenberg has taught a selector coupled to an output of the instruction cache system and to an output of the segment cache (Rotenberg Section 2.1 Trace-Level Sequencing "…The output of the trace cache is one or more traces…"; Section 2.2 Instruction-Level Sequencing "The *outstanding trace buffers* in Fig. 2 are used to 1) construct new traces that are not in the trace cache and 2) track branch outcomes…"; and Figure 2 – In regards to Rotenberg, Figure 2 shows the output of the trace cache and the outstanding trace buffers are connected to a line that has two inputs and one output, which is a selector. The selector chooses between and existing trace in the trace cache, i.e. a trace cache hit, or a newly formed trace, i.e. a trace cache miss or misprediction.) A person of ordinary skill in the art at the time the invention was made, and as taught by Rotenberg, would have recognized that the trace cache system provides fast trace-level sequencing while providing a method to create nonexistent traces or repair mispredicted traces (Rotenberg Section 2.1 Trace-Level Sequencing "…The trace predictor and trace cache together provide fast trace-level sequencing…Instruction-level sequencing, discussed in the next section, is required to construct nonexistent traces or repair trace mispredictions."). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the trace caches, buffers, and selector of Rotenberg in the device of Peleg to increase the speed of trace-level sequencing while using instruction-level sequencing to create new or correct mispredicted traces.

28. Referring to claim 16, Peleg in view of Rotenberg has taught the processing engine of claim 15, wherein the instruction segments are extended blocks (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

29. Referring to claim 17, Peleg in view of Rotenberg has taught the processing engine of claim 15, wherein the instruction segments are traces (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

30. Referring to claim 18, Peleg in view of Rotenberg has taught the processing engine of claim 15, wherein the instruction segments are basic blocks (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

31. Referring to claim 19, Peleg in view of Rotenberg has taught the processing engine of claim 15, wherein the instruction segment cache system further comprises a segment predictor provided in communication with the segment cache.

32. Claims 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peleg et al., U.S. Patent Number 5,381,533 (herein referred to as Peleg) as applied to claim 8 above, and further in view of Kyker et al., U.S. Patent Number 6,578,138 (herein referred to as Kyker).

33. Referring to claim 9, Peleg has taught the method of claim 8, further comprising building a second instruction segment based on program flow (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12). Peleg has not taught if

the first and second instruction segments overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment. Kyker has taught if the first and second instruction segments, overlap, extending the first instruction segment to include non-overlapping instructions from the second instruction segment (Kyker Abstract "…a cache unit, which includes a data array that stores traces…In one exemplary method of unrolling loops, the processor or trace cache unrolls loops…"; column 2, line 59 to column 3, line 33 "…when the trace cache determines that a loop is present, the trace cache continues to build the trace by building additional iterations of the loop until the trace is a minimal length…In other words, the trace cache builds the loop repeatedly until the trace is, for example, over two trace lines long…"; Figure 1; and Figure 2 – In regards to Kyker, as shown in Figure 1, the trace is built as normal, e.g. a first instruction segment is built normally, when there is no backward-taken branch. When a backward branch is taken to form another, second instruction segment, another iteration of the loop is added to the trace, thereby extending the first segment to include the first segment.). A person of ordinary skill in the art at the time the invention was made would have recognized that unrolling loops for a single continuous block of instructions improves processor efficiency by reducing the number of times the processor accesses the cache and maximizing usage of a cache line. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the loop unrolling of Kyker in the device of Peleg to improve efficiency.

34. Referring to claim 10, Peleg in view of Kyker has taught the method of claim 9, wherein the extending comprises storing the non-overlapping instructions in the cache in reverse program order in successive cache positions adjacent to the instructions from the first instruction segment

(Kyker Abstract "…a cache unit, which includes a data array that stores traces…In one exemplary method of unrolling loops, the processor or trace cache unrolls loops…"; column 2, line 59 to column 3, line 33 "…when the trace cache determines that a loop is present, the trace cache continues to build the trace by building additional iterations of the loop until the trace is a minimal length…In other words, the trace cache builds the loop repeatedly until the trace is, for example, over two trace lines long…"; Figure 1; and Figure 2 – In regards to Kyker, as shown in Figure 2 and explained the in the cited lines, the loop has three main instructions and, when the backwards branch is encountered, the program jumps backwards, i.e. reverses directions, to the beginning of the loop instead of moving forward in the program. As shown in Figure 2, when L_H is encountered, previously stored instructions L_2 and L_3 are stored again in consecutive lines in the cache, e.g. the trace cache reverses the program direction and stores previously stored instructions again in the trace cache when a loop is encountered.).

Conclusion

35. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

36. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

37. Any inquiry concerning this communication or earlier communications from the examiner should be directed to AIMEE J. LI whose telephone number is (571)272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

38. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

39. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aimee J Li/
Primary Examiner, Art Unit 2183

16 August 2009